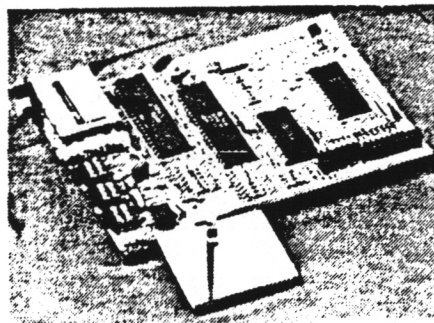
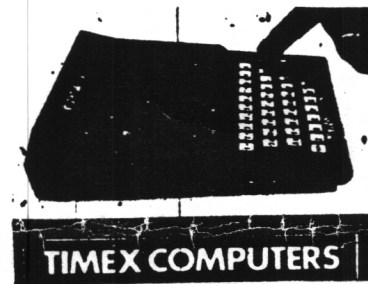


LISTing Newsletter

Newsletter of the Long Island
Sinclair/Timex Users Group

Next Meeting

MAY 7, 1995



Game
Program
Issue

Listing Policy

Annual Dues \$16.00

One "sample" copy sent upon receipt of Business size SASE.
Copies provided on EXCHANGE BASIS with other bona fide user
groups. LISTing is published monthly except July and August by
LIST (Long Island Sinclair Timex) Group, a not for profit user group.

We are always looking for articles, programs, reviews etc. to keep our
members informed and entertained. You maintain full credit and copyright.

Portions if this publication may be reproduced, without written consent.
Please give credit to LIST when reprinting articles.

LIST disclaims any responsibility for any damage you may do to your
computer as a result of reading any articles in LISTing.

L.I.S.T. OFFICERS

PRES. HARVEY RAIT
VICE PRES. BOB GILDER
COR.SECY. JOHN PAZMINO
ASSOC EDITORS FRED STERN
HARVEY RAIT
PUBLISHER BOB GILDER
LIBRARIAN TOM SKAPINSKY
TREASURER BOB MALLOY

PLEASE SEND ALL INQUIRIES AND
SUBMISSIONS(INCLUDING DUES)
TO; L.I.S.T.

HARVEY RAIT
5 PERI LANE
VALLEY STREAM,N.Y.11581

COMING EVENTS:THE NEXT L.I.S.T.
MEETING WILL BE SUNDAY MAY 7
AT 2 P.M.AT THE HOME OF HARVEY
RAIT (SEE ADDRESS ABOVE).

REPORT ON THE MEETING OF APRIL
9:

THE MEETING STARTED ON TIME AT
2:00 P.M. WITH 4 MEMBERS PRES-
ENT.

ONE NEW MEMBERSHIP CAME IN AND
WE TAKE THIS OPPORTUNITY TO
WELCOME KENNETH HARBIT OF
FRESNO CALIFORNIA.

ONE INQUIRY WAS RECIEVED FROM
ERIC HILL OF MILWAUKEE
WISCONSIN,WHOSE QUESTIONS WILL
BE ANSWERED.

WE DECIDED TO KEEP THE MEETING
INFORMAL AND WOUND UP HAVING A
ROUNDTABLE DISCUSSION OF
VARIOUS TOPICS INCLUDING THE
OBVIOUS MERITS OF THE SINCLAIR
IN COMPARISON TO THE CURRENT
PLETHORA OF SO CALLED ADVANCES
IN THE COMPUTER WORLD TODAY.
IT TURNS OUT THAT SO MANY OF
THE CURRENT BALLYHOOD ADVANCES
IN THE PC WORLD TODAY ARE
MERELY VARIATIONS OF THINGS
THAT WE HAVE BEEN DOING FOR
YEARS WITH OUR TOYS.

BOB GILDER GAVE A DEMONSTRATION
OF THE "EDITOR" PROGRAM TO SHOW
HOW TO PRINT OUT A TWO COLUMN
PAGE IN ONE PASS SIMILAR TO
THIS PAGE.IT WAS A LITTLE
COMPLICATED TO A NOVICE THAT
NEVER USED THIS PROGRAM
BEFORE,BUT BOB WHO USES IT ALL
THE TIME MADE IT LOOK EASY.
MEETING ADJOURNED AT 4:15.

FIRST OFF I WANT TO REPEAT THAT THE
NEXT MEETING WILL BE ON MAY 7,WHICH
IS THE FIRST SUNDAY OF THE MONTH
INSTEAD OF THE SECOND SUNDAY BECAUSE
OF MOTHERS DAY FALLING ON THE 14 th.

IN GOING THROUGH MY "OLD STUFF" I
CAME ACROSS A LABORATORY EXPERIMENT
FROM MY COLLEGE DAYS WHICH WAS MY
FIRST EXPOSURE TO COMPUTERS.

THE COOPER UNION DEPT. OF ELECTRICAL
ENGINEERING,DATED 3/27/1961.TOPIC -
ELECTRONIC ANALOGUE COMPUTER.

THE PARAMETERS WERE SET UP IN
INCREMENTAL STEPS TO PRODUCE RESULTS
ON A STRIP CHART RECORDER.THE
PROBLEM CHOSEN WAS TO ANALYZE THE
EFFECT OF THE DAMPING COEFFICIENT OF
AN AUTOMOBILE'S SHOCK ABSORBER ON
THE AUTOMOBILES TRANSLATIONAL MOTION
WHEN RIDING OVER A RECURRENT BUMP.
THE PARAMETERS WERE THAT THE BUMP
WILL BE EQUIVALENT TO A DISPLACEMENT
OF 0.2 ft AND LAST LAST FOR A
DURATION OF 0.01 sec.

DIFFERENTIAL EQUATIONS WERE SET UP
USING SUCH PARAMETERS AS THE MASS OF
THE AUTO,THE MASS OF THE TIRE AND
AXLE,THE LINEARIZED SPRING CONSTANT
OF A PNEUMATIC TIRE,AND THE
CONSTANTS ASSOCIATED WITH THE SHOCK
ABSORBER DAMPING CONSTANT 'C1'.
CALIBRATED AMPLIFIERS WERE SET UP
AND AN INPUT SIGNAL APPLIED.OUTPUTS
WERE FED TO A STRIP RECORDER AND
ANALYZED.

ALL IN ALL THIS WAS A COMPLICATED
SET-UP TO OBTAIN RESULTS.

WHY? YOU MAY ASK AM I WRITING ABOUT
THIS.I GUESS YOU WOULD HAVE HAD TO
BE AROUND IN THE 50'S AND EARLY 60'S
TO FULLY APPRECIATE THE BIRTH AND
MATURATION OF THE DIGITAL COMPUTER
AS WE NOW USE IT.THE 2K TS-1000
COULD HAVE SOLVED THIS PROBLEM FAR
MORE EFFICIENTLY AND EXPEDITIOUSLY
THAN THE HOURS OF PREPARATION
REQUIRED TO SOLVE THE STATED PROBLEM
THAT WERE REQUIRED WITH AN ANALOGUE
COMPUTER.

A 16 YEAR OLD HACKER CAN NOW DO WHAT
ONCE REQUIRED YEARS OF STUDIES IN
PHYSICS AND DIFFERENTIAL ALGEBRA CAN
WE REALLY LONG FOR THE MORE SIMPLE
GOOD OLD DAYS?I DONT KNOW THE ANSWER
BUT I STILL KIND OF LOOK BACK WITH
NOSTALGIA AND THINK OF CAR RADIOS
THAT USED A VIBRATOR AND AN OZ4 TO
GENERATE DC VOLTS.

BY THE WAY I DID GET AN "A" IN THE
COURSE.

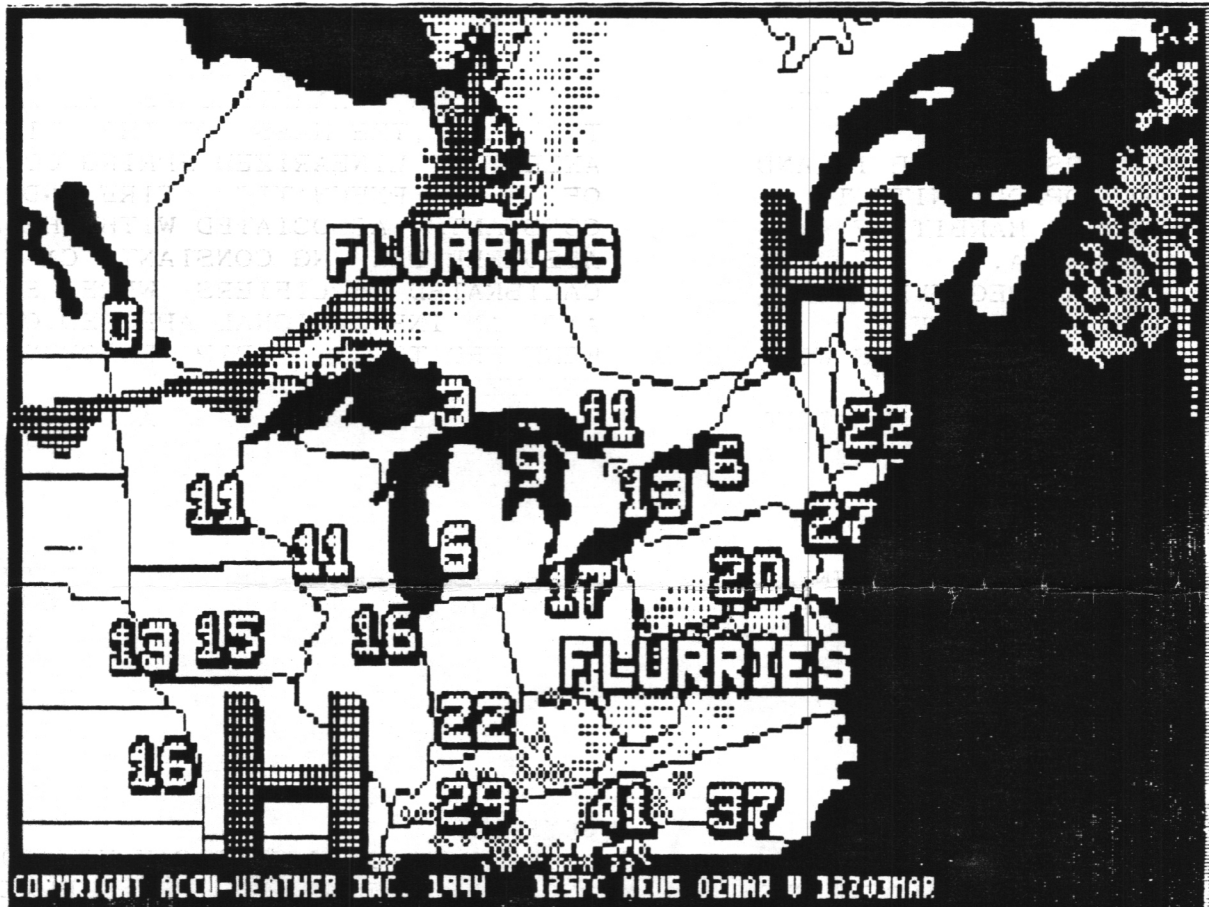
QL CORNER

During the past two months I have been reviewing some of the IQLR Freeware, Public Domain, Shareware Software which I had ordered from Bob Dyl at \$2.50 each. IQLR-013, UNGIF allows you to review and save onto a QL diskette, GIF format screens which can be located on many bulletin boards which we can download into our QL's. The software disk contains several weather maps in the GIF format that you can load into the UNGIF utility.

The ENGif utility converts partial or complete QL screens to GIF files.

The UNGIF utility requires several Pointer Interface files for this utility which ARE NOT supplied; HOT_REXT, MENU_REXT, WMAN and PTR_GEN. A five page manual accompanies the UNGIF utility and a two page manual for the ENGIF utility. Both manual files have a 'txt' extension. If you don't have a text editor you can import the files into Quill.

To use the Ungif utility 'lrun' the BOOT file which is chained to BOOT_UNgif. The Ungif_BOOT stops at line 50, the last line in this file. Line 50: 50 EX_ungif; drive_name. Delete the drive_name and enter FLPL_ (or any other drive designator) Neast2_gif (one of the gif files supplied with the program disk). Run the BOOT_UNgif. Within several seconds the weather map for the North East section of the USA appears (I believe). See the screen dump of this map.



At this time, you should depress either the ESC, SPACE or ENTER key to enter the Main menu. (See the Main Menu screen dump). There are ten (10) Main Menu functions which can be activated with UP/DOWN cursor keys, pressing a letter at the right side of each function or a mouse if you have the mouse interface or serial mouse.

The first function on the menu, QL screen mode : 4-colour. Pressing Space will toggle this line for 8-colour and again monochrome. After setting this function to another mode, say 8-colour, press 'R' (redraw

GIF) and the screen will be redrawn with the 8-colour tones. Experiment with each of the three modes for the best picture display.

The next function, T - Type of picture: monochrome can be changed to color or greytone by pressing the Space bar or a 'Mit'. Again, experiment, using each of these modes by redrawing the picture.

The third function L - Stipple level : 0 = 1x1 can be changed to five different levels including 'dither' which provides 9 greytones, 64 intensity levels or 1000 different stipple-levels in mode 8. Dither provides images of excellent quality and smoothness. It's really a fuzzy method to render a large number of intensities.

V - Save-file type : SCR screen: Toggling the space bar will set the screen extension to SCR, GIF or PIC.

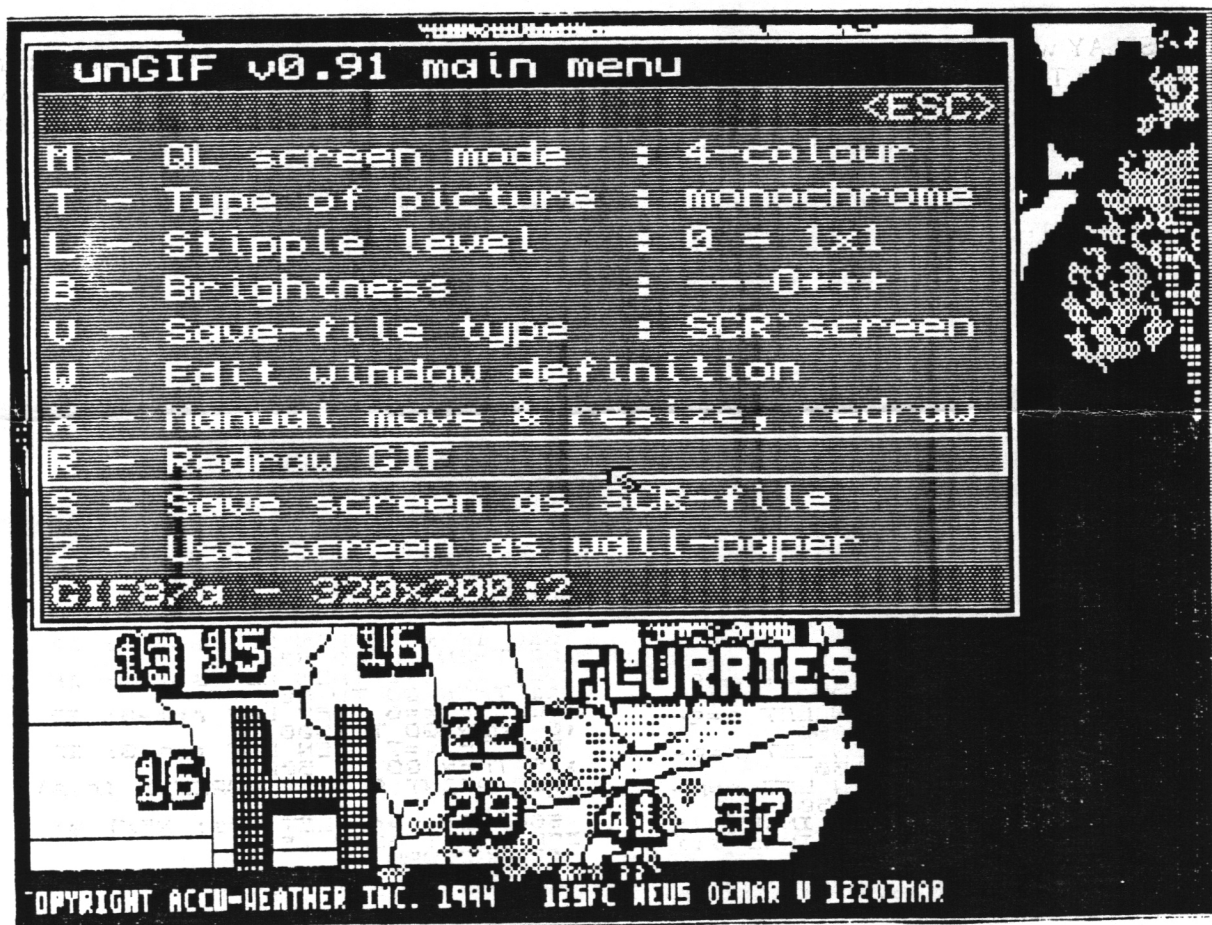
W - Edit the window defination: must be a CON_ device. CON_300 becomes 300x180a32x16.

X - Manual move and resize, redraw: Using the cursor keys (or mouse) will allow the screen size to be altered, then redrawn to the new size.

S - Save screen as SCR-file: when this function is activated, delete the screen name as appears and enter a new name and the 'scr' extension.

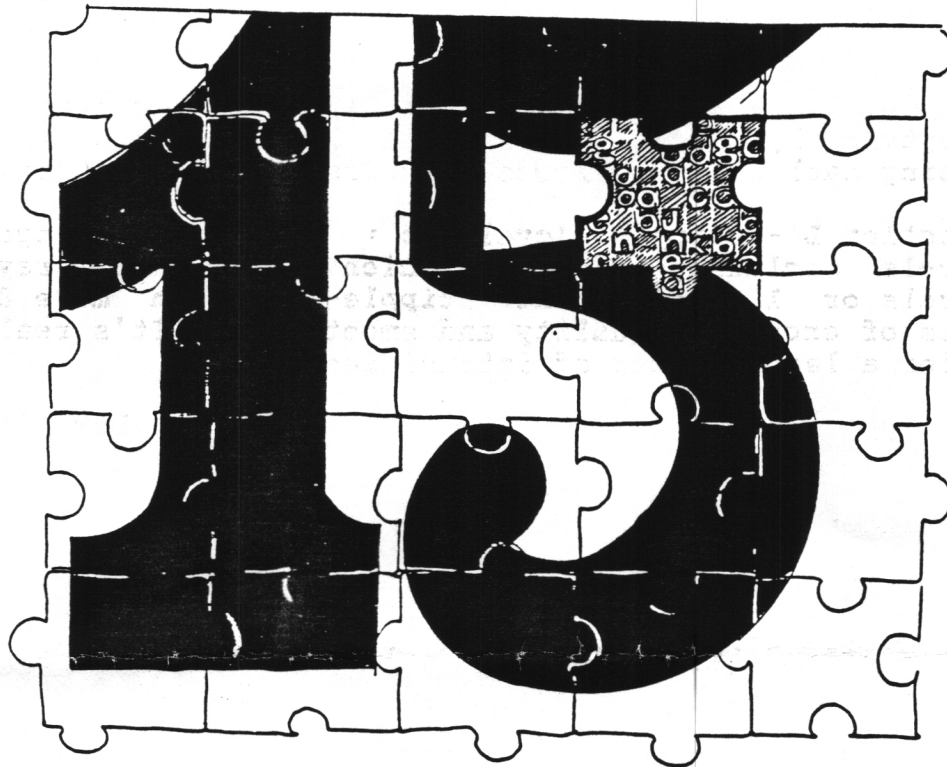
Z - Use screen as wall-paper: I have not yet used this function!

So, there you have it! If you are into developing a large screen shot library, then this program can be a blessing for you. At \$2.50 for the program, it's a steal.



See you next month... Bob Gilder

TS1000
TS2068



Puzzle

THE DISPLAY will show a grid of 16 squares. The grid contains the first 15 letters of the alphabet and one space. Slide the letters around in the grid, using the space,

and the puzzle is solved when the letters are in their correct order.

The letters are moved with the usual cursor controls and a count is kept of the moves taken. Your task is

to complete the re-arrangement in as few moves as possible. A diverting little routine, worth having on tape. Submitted by W G Davies of Hereford for the 16K Spectrum.

```

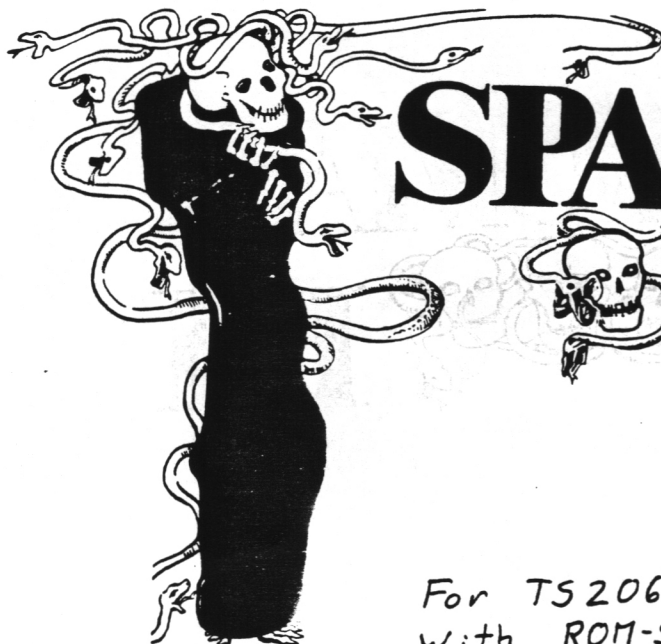
10 REM "15 PUZZLE"
20 REM © W G Davies 1982
100 PRINT AT 8,11:"15 PUZZLE":/
T 20,6:"Choose your colours"
110 LET a$="QEMKLUOGAHFDNIR "
120 LET l=1000: DIM b$(16)
130 LET c$="ABCDEFGHIJKLMNO "
140 INPUT "Border ";a,"Ink ";b
150 BORDER a: INK b: PAPER 7
170 CLS: FOR c=80 TO 176 STEP
24
180 PLOT c,40: DRAW 0,96: NEXT
c
190 FOR d=40 TO 136 STEP 24: PL
OT 80,d: DRAW 96,0: NEXT d
200 LET j=0: LET k=1: LET p=15:
LET q=20
220 FOR n=6 TO 15 STEP 3: FOR m
=11 TO 20 STEP 3: PRINT AT n,m;2
$(k)
230 LET k=k+1: NEXT m: NEXT n
240 GO SUB 490
250 LET r=p: LET s=q: IF INKEY$
=" " THEN GO TO 250
260 IF INKEY$="5" THEN LET s=q+
3: IF s>20 THEN GO TO 250
270 IF INKEY$="8" THEN LET s=q-
3: IF s<11 THEN GO TO 250
280 IF INKEY$="6" THEN LET r=p-
3: IF r<6 THEN GO TO 250
290 IF INKEY$="7" THEN LET r=p+
3: IF r>15 THEN GO TO 250
300 IF INKEY$<>"5" AND INKEY$<>
"8" AND INKEY$<>"7" AND INKEY$<>
"6" THEN GO TO 250
310 PAPER 7: BEEP .1,0: PRINT A
T p,q:SCREEN$(r,s):AT r,s:" "

```

```

320 GO SUB 400: LET p=r: LET q=
s
330 LET j=j+1: PRINT AT 3,15;j
350 GO TO 250
400 LET a=1: FOR n=6 TO 15 STEP
3: FOR m=11 TO 20 STEP 3: LET b
$(a)=SCREEN$(n,m)
410 LET a=a+1: NEXT m: NEXT n
420 IF b$=c$ THEN GO TO 440
430 RETURN
440 PRINT AT 19,12:"CORRECT": F
OR b=1 TO 16: BEEP .5,b: NEXT b
450 PRINT AT 21,2:"Press any ke
y to play again."
460 IF INKEY$="" THEN GO TO 460
470 PRINT AT 21,2:"
480 PRINT AT 19,12:" "
490 SUB 600: GO TO 200
500 FOR y=1 TO 50
510 LET r=p: LET s=q: LET x=INT
(RND*4)+5
520 IF x=5 THEN LET s=q+3: IF s
>20 THEN GO TO 500
530 IF x=8 THEN LET s=q-3: IF s
<11 THEN GO TO 500
540 IF x=6 THEN LET r=p-3: IF r
<6 THEN GO TO 500
550 IF x=7 THEN LET r=p+3: IF r
>15 THEN GO TO 500
560 PRINT AT p,q:SCREEN$(r,s):
AT r,s:" "
570 LET p=r: LET q=s: NEXT y: B
EEP .5,9: BEEP .5,5
580 RETURN
600 IF j<1 THEN LET l=j
610 PRINT AT 3,15;"0 " AT 1,
5:"Best score so far:";l;" "
RETURN

```



SPACMAN III

For TS2068
With ROM-SWITCH



WE FEEL as if we are lost in a trackless maze, pursued by vengeful ghosts and forced to consume endless quantities of your Spaceman programs — but we like it.

The latest breakthrough is from Duncan Anderson of Bishops Frome, Worcester, who has managed to incorporate power pills and intelligent ghosts, which pursue you alternately with carnivorous cunning or flee from you with the agility born of terror. The maze is suitably spaghetti-like, Mr Pacman's jaws face forward, there are exits, and a novel feature is the bomb you can leave behind you if the ghost's icy breath is too close on your collar.

You have only one life in **Spaceman III** and there are only two ghosts which do not return to the centre when killed. An additional problem is that the ghost's direction finder becomes confused by corners. A possible way round the problem is to use only straight lines to form the walls of the maze, as do some of the commercial versions of the game.

We would say this listing is a big leap forward and we feel sure that there must be some Clive Sinclair clone who can put all the pieces together and supply us with the real thing (48K Spectrum). Graphics notes:

88,540 etc—Graphic R.
110—Dots are graphic.
420, etc—Graphic P.
510, etc—Graphic L.
520, etc—Graphic D.
530, etc—Graphic U.
2740, etc—Graphic G.

```

5 LET 1020=0
10 RESTORE : DATA 0,0,0,BIN 00
011000,BIN 00011000,0,0,0,BIN
1000010,BIN 11100111,255,255,BIN
1111110,BIN 111100,BIN 11000,BI
N 000011100,BIN 111110,BIN 11111
00,BIN 111000,BIN 111000,BIN 1
111100,BIN 111110,BIN 11100
12 DATA 0,BIN 00011000,BIN 001
11100,BIN 01111110,255,255,BIN 1
1100111,BIN 0100010
14 DATA BIN 00111000,BIN 01111
100,BIN 00111110,BIN 00011110,BI
N 00011110,BIN 00111110,BIN 0111
1100,BIN 00111000
16 DATA 01111110,BIN 10010011,BI
N 10010011,255,255,BIN 11010101
18 DATA 0,BIN 01111110,BIN 010
0010,BIN 01011010,BIN 01011010,
BIN 0100010,BIN 01111110,0
20 FOR n=0 TO 7: READ x: POKE
USR "0"+n,x: NEXT n
30 FOR n=0 TO 7: READ x: POKE
USR "U"+n,x: NEXT n
40 FOR n=0 TO 7: READ x: POKE
USR "4"+n,x: NEXT n
50 FOR n=0 TO 7: READ x: POKE
USR "4"+n,x: NEXT n
60 FOR n=0 TO 7: READ x: POKE
USR "1"+n,x: NEXT n
70 FOR n=0 TO 7: READ x: POKE
USR "3"+n,x: NEXT n
80 FOR n=0 TO 7: READ x: POKE
USR "P"+n,x: NEXT n
25 GO SUB 9000
88 LET b0="C": LET sc=0: LET b
on=0: LET x=20: LET y=8: LET gx1
=10: LET gx2=10: LET gy1=11: LET
gy2=21
90 DIM b$(21,31)
100 LET b$(1)=
110 LET b$(2)=
120 LET b$(3)=
130 LET b$(4)=
140 LET b$(5)=
150 LET b$(6)=
160 LET b$(7)=
170 LET b$(8)=
180 LET b$(9)=
190 LET b$(10)=
200 LET b$(11)=
210 LET b$(12)=
220 LET b$(13)=
230 LET b$(14)=
240 LET b$(15)=
250 LET b$(16)=
260 LET b$(17)=
270 LET b$(18)=
280 LET b$(19)=
290 LET b$(20)=
300 LET b$(21)=
340 LET b1=0: LET b4=999
350 LET pp=0
355 LET gx=0

```



SPACMAN III



```
350 LET b=0
370 LET PPX=INT (RND*20+1): LET
PPY=INT (RND*30+1)
380 IF b$(PPX,PPY)="" THEN GO
TO 370
400 BORDER 1: PAPER 7: INK 1: C
LS
410 PRINT " "
FOR n=1 TO 21: PR
INT "b$(n): NEXT n
420 PRINT AT 0,0: PAPER 1: INK 2: "B"
AT 10,1: INK 6: PAPER 1: "B"
AT 10,31: INK 6: PAPER 1: "B"
500 IF INKEY$="" THEN GO TO 550
510 IF INKEY$="S" THEN LET b$="
520 IF INKEY$="6" THEN LET b$="
530 IF INKEY$="7" THEN LET b$="
540 IF INKEY$="8" THEN LET b$="
545 IF b=0 AND INKEY$="0" THEN
GO SUB 7000
550 PRINT AT x,y: " ": LET b$(x,
y)=
560 IF b$=" " AND b$(x-1,y)=""
THEN LET x=x-1
570 IF b$="A" AND b$(x+1,y)=""
THEN LET x=x+1
580 IF b$="6" AND b$(x,y+1)=""
THEN LET y=y+1
585 IF b$="7" AND b$(x,y-1)=""
THEN LET y=y-1
590 PRINT INK 0: AT x,y: b$
600 IF b$(x,y)="" THEN LET sc=
sc+1: BEEP .015,5: PRINT AT 0,2:
sc+b0n
610 IF sc=342 OR sc=784 OR sc=1
226 OR sc=1568 THEN GO TO 4000
620 IF 9x1=9x2 AND 9y1=9y2 THEN
GO SUB 9000
635 PRINT AT 9x2,9y2: b$(9x2,9y2
)
640 PRINT AT 9x1,9y1: b$(9x1,9y1
)
641 IF RND<.2 THEN FOR n=0 TO 1
0: NEXT n: GO TO 700
650 IF b$(9x1+1,9y1)="" AND 9
x1<x THEN LET 9x1=9x1+1
655 IF b$(9x2+1,9y2)="" AND 9
x2<x THEN LET 9x2=9x2+1
660 IF b$(9x1-1,9y1)="" AND 9
x1>x THEN LET 9x1=9x1-1
665 IF b$(9x2-1,9y2)="" AND 9
x2>x THEN LET 9x2=9x2-1
670 IF b$(9x1,9y1+1)="" AND 9
y1<y THEN LET 9y1=9y1+1
675 IF b$(9x2,9y2+1)="" AND 9
y2<y THEN LET 9y2=9y2+1
680 IF b$(9x1,9y1-1)="" AND 9
y1>y THEN LET 9y1=9y1-1
685 IF b$(9x2,9y2-1)="" AND 9
y2>y THEN LET 9y2=9y2-1
740 PRINT AT 9x1,9y2: INK 3: "B"
745 PRINT AT 9x1,9y2: INK 3: "B"
746 IF PP=0 THEN PRINT AT PPX,PP
Y: INK 2: "B"
748 IF b=1 THEN PRINT AT bx,by:
INK 4: "B"
750 IF (9x1=x AND 9y1=y) OR (9x
2=x AND 9y2=y) THEN GO TO 1000
760 PRINT AT 0,2: sc+b0n
765 IF ex=5 THEN GO TO 780
770 IF x=10 AND y=2 AND b$="3"
THEN PRINT AT x,y: " ": BORDER 6:
BEEP .1,24: LET y=30: BORDER 1:
GO TO 779
775 IF x=10 AND y=30 AND b$="6"
THEN PRINT AT x,y: " ": BORDER 6:
BEEP .1,24: LET y=2: BORDER 1:
GO TO 779
779 GO TO 780
779 LET ex=ex+1: IF ex>5 THEN P
```

```
PRINT AT 10,3: " " AT 10,31: " "
780 IF x=PPX AND y=PPY AND PP=0
THEN LET PP=1: GO SUB 2000
790 IF b=1 AND ((9x1=bx AND 9y1
=by) OR (9x2=bx AND 9y2=by)) THE
N GO SUB 8000
8000 GO TO 500
8000 PRINT AT x,y: INK 0: PAPER
6: FLASH 1,25: FOR n=24 TO -24 3
TEP -1: BEEP .05,n: NEXT n
1002 BORDER 0: PAPER 0: INK 6: C
LS: PRINT AT 10,11: FLASH 1,"SC
ORE": sc+b0n
1004 FOR n=0 TO 100: BEEP .01,n:
50: NEXT n
1005 LET l090=1: GO TO 0
1000 LET b0n=b0n+5: BORDER 2: FO
R n=10 TO 20: BEEP .05,n: NEXT n
2100 FOR y=0 TO 20
2500 IF INKEY$="" THEN GO TO 255
2510 IF INKEY$="5" THEN LET b$="
2520 IF INKEY$="6" THEN LET b$="
2530 IF INKEY$="7" THEN LET b$="
2540 IF INKEY$="8" THEN LET b$="
2550 PRINT AT x,y: " ": LET b$(x,
y)=
2560 IF b$=" " AND b$(x-1,y)=""
THEN LET x=x-1
2570 IF b$="A" AND b$(x+1,y)=""
THEN LET x=x+1
2580 IF b$="6" AND b$(x,y+1)=""
THEN LET y=y+1
2585 IF b$="7" AND b$(x,y-1)=""
THEN LET y=y-1
2590 PRINT AT x,y: INK 0: b$
2600 IF b$(x,y)="" THEN LET sc=
sc+1: BEEP .01,20: PRINT AT 0,2:
sc+b0n
2635 PRINT AT 9x2,9y2: b$(9x2,9y2
)
2640 PRINT AT 9x1,9y1: b$(9x1,9y1
)
2641 IF RND<.3 THEN FOR n=0 TO 1
0: NEXT n: GO TO 2700
2650 IF b$(9x1+1,9y1)="" AND 9
x1<x THEN LET 9x1=9x1+1
2655 IF b$(9x2+1,9y2)="" AND 9
x2<x THEN LET 9x2=9x2+1
2660 IF b$(9x1-1,9y1)="" AND 9
x1>x THEN LET 9x1=9x1-1
2665 IF b$(9x2-1,9y2)="" AND 9
x2>x THEN LET 9x2=9x2-1
2670 IF b$(9x1,9y1+1)="" AND 9
y1<y THEN LET 9y1=9y1+1
2675 IF b$(9x2,9y2+1)="" AND 9
y2<y THEN LET 9y2=9y2+1
2680 IF b$(9x1,9y1-1)="" AND 9
y1>y THEN LET 9y1=9y1-1
2685 IF b$(9x2,9y2-1)="" AND 9
y2>y THEN LET 9y2=9y2-1
2740 PRINT AT 9x1,9y1: INK 0: "B"
2745 PRINT AT 9x2,9y2: INK 0: "B"
2750 IF (9x1=x AND 9y1=y) OR (9x
2=x AND 9y2=y) THEN GO TO 3000
2770 IF x=10 AND y=2 AND b$="3"
THEN PRINT AT x,y: " ": BEEP .01,
24: LET y=30
2710 IF x=10 AND y=30 AND b$="6"
THEN PRINT AT x,y: " ": BEEP .05
,24: LET y=2
2800 NEXT y
2900 BORDER 0: FOR n=0 TO 3: BEE
P .1,0: BEEP .1,1: NEXT n: BORDE
R 1
2910 RETURN
3000 PRINT AT x,y: BRIGHT 1: FLA
SH 1: INK 2: PAPER 6
3010 FOR a=1 TO 3: FOR n=12 TO 2
```

```
4: BEEP .05,n: NEXT n: NEXT a
3020 LET b0n=b0n+25: PRINT AT 0,
2: sc+b0n
3021 PRINT AT 9x1,9y1: b$(9x1,9y1
): AT 9x2,9y2: b$(9x2,9y2)
3022 LET 9x1=INT (RND*20+1): LET
9y1=INT (RND*30+1): IF b$(9x1,9
y1)="" THEN GO TO 3022
3026 LET 9x2=INT (RND*20+1): LET
9y2=INT (RND*30+1): IF b$(9x2,9
y2)="" THEN GO TO 3026
3028 PRINT AT 9x1,9y1: INK 3: "A"
AT 9x2,9y2: INK 2: "A"
3030 RETURN
4000 LET sc=sc+100: BORDER 6: PA
PER 0: INK 3: CLS
4010 PRINT AT 10,30: INK 2: "B"
4020 FOR x=1 TO 27: PRINT AT 10,
x: INK 3: "B": AT 10,x+3: INK 0:
" ": BEEP .05,n: NEXT x
4030 FOR n=12 TO 24: BEEP .01,n:
NEXT n
4040 FOR j=27 TO 1 STEP -1: PRIN
T AT 10,j: INK 0: "A": AT 10,j+3:
INK 2: "A": BEEP .05,50-j: NEXT
j
4050 GO TO 100
7000 BORDER 4: LET bx=x: LET by=
y: FOR n=0 TO 20: BEEP .01,n: NE
XT n: PRINT AT bx,by: INK 4: "B"
LET b=1: BORDER 1
7100 RETURN
8000 LET b=2: PRINT AT bx,by: FL
ASH 1: INK 4: PAPER 0: "A": FOR n
=0 TO 40: BEEP .01,n: NEXT n: PR
INT AT bx,by: b$(bx,by): LET b0n=
b0n+15
8100 PRINT AT 9x1,9y1: b$(9x1,9y1
): AT 9x2,9y2: b$(9x2,9y2)
8100 LET 9x1=INT (RND*20+1): LET
9y1=INT (RND*30+1): IF b$(9x1,9
y1)="" THEN GO TO 8100
8100 LET 9x2=INT (RND*20+1): LET
9y2=INT (RND*30+1): IF b$(9x2,9
y2)="" THEN GO TO 8100
8200 RETURN
9000 LET 9x1=20-x: LET 9y1=30-y
9005 IF y=1 THEN GO SUB 9000
9010 IF b$(9x1,9y1)="" THEN LET
9y1=9y1-1: GO TO 9005
9000 LET x=x+1
9010 IF x>10 THEN LET x=1
9020 IF b$(9x1,9y1)="" THEN RE
TURN
9040 RETURN
9050 IF b$=" " SPACMAN 3 BY DUNC
AN ANDERSON
9060 BORDER 0: PAPER 0: INK 6: C
LS
9080 PRINT AT 10,0: "PRESS ANY KE
Y": PAPER 0
9085 IF l090=1 THEN RETURN
9010 FOR n=1 TO 31: BEEP .01,n:
LET b$=b$(1 TO n): PRINT AT 10,3
1: LEN b$: b$: NEXT n
9015 FOR n=0 TO 30: PRINT AT 10,
n: " ": AT 10,n+1: INK 3: BRIGHT 1
n: " ": BEEP .1,10+n: NEXT n
9030 PRINT AT 10,31: "INSTRUCTIONS:"
9040 PRINT AT 2,0: " "
AT 5,0: INK 7: "A": AT 7,5: INK 3
"A": AT 9,6: INK 2: "B": AT 11,6:
INK 6: "B": AT 13,6: INK 4: "B"
9050 FOR n=0 TO 30: PRINT AT 5+2+
n,0: " "
9055 PRINT AT 0,14: "YOU": AT 7,14
"HOST": AT 9,14: "POWER PILL": AT
11,14: "EXIT": AT 13,14: "BOMB (PR
ESS 5)":
9070 FOR n=10 TO 40: BEEP .1,n:
NEXT n
9080 RETURN
```


TS 1000



THE INVISIBLE TOOLKIT is a suite of fast efficient machine-code routines driven by a short Basic menu. Due to the fact that extensive use is made of routines already present in the Rom the machine-code is written in less than 800 bytes, which together with the Basic lines occupies less than 1800 bytes of memory.

The program was written to make it easier to modify and develop a Basic program and even combine two or more Basic programs — or routines — so that the idea of having a library of useful subroutines on cassette is quite feasible. In addition, the facility to create large Rems, or delete them at a stroke, is invaluable when assembling machine-code.

The following routines are available:

DOWN CODE — used after Up Code. Downloads the copy in upper memory into a specified line and Address. Space is automatically created.

DOWN BASIC — used after Up Basic. Brings the Basic copy down into the Basic area placing it according to the first line number of the block.

LINE ADDRESS — gives the address of the specified line number. The first character is four bytes further on.

REM EXPAND — expands the Rem statement with the specified line number, from a specified address, by a given number of bytes.

UP CODE — copies a given number of bytes from a given address, into upper memory.

THE INVISIBLE TOOLKIT

Stuart Clarke presents a number of machine-code routines for the ZX-81 which are driven by a short Basic menu.

UP BASIC — copies a block of Basic lines specified by first and last lines.

BLOCK DELETE — specify first and last lines.

RENUMBER BASIC — renumbers a block of Basic lines specified by first and last lines. You are allowed to specify both the Increment and the New first line number. All non-computed Gotos, Gosubs, and Runs will be correctly changed, provided the line number is in four-digit form, e.g., Goto 1 as Goto 0001.

RENUMBER COPY — used after Up Basic. Only requires the Increment and New first line number to be specified.

The order in which the routines are listed may seem a little odd, but this is as they appear in the menu. This is dictated by the use of relative rather than absolute jumps in the machine-code to allow it to be position independent, and the entire program has been written in such a way that, as well as the machine code, the Basic menu can be transferred to upper memory — without overwriting existing upper memory contents — to allow the Loading of subsequent programs.

When required, the menu can be recalled, and it will not clash with the existing Basic as its lines are numbered A222 to A239 — 10222 to 10239. This is possible since the Rom checks the validity of a line number by only comparing the high byte with 40 decimal — 40 times 256 equals 10240, the first "non-valid" line number.

However, line numbers in the range 10000 to 11239 are only obtained by using the Renumber routine, as 9999 is the largest line you can enter direct from the keyboard. The menu lines are now tagged on to the end of the program you have just Loaded.

Report codes.

K/A234 you have attempted to use Down Code when no copy exists in upper memory.

T/A233 you have attempted to use Down Basic or Renumber Basic when the "live" copy is not Basic.

System variables.

16388-16389 Ramtop
16507-16508 Old Ramtop
16444-16451 Poked by menu
ZX81 variable pointing to the first location in the "jump table" accessing the machine-code routines.

If the typing seems a bit of a chore, just send your name and address with a cheque for £3 made payable to Sarglen Software, to 147 South Parks Road, Glenrothes, Fife, KY6 1NT, and I will send you a quality cassette with two copies each for machines with and without the "improved Rom", together with an instruction leaflet.

Type in program 1, then duplicate line 1 using the Edit key so that you have 12 identical lines, numbered 1 to 12. Now Run program 1.

Type in program 2, and Run it. This is the hex loader. Enter the address 16544 — or the

Program 1.

```
1 REM
.....
20 POKE 16511,68
```

```
30 POKE 16512,3
40 FOR N=16514 TO 17347
50 POKE N,1
60 NEXT N
```

Program 2.

```
10 INPUT S
20 LET A$=""
30 GOSUB 0150
40 IF A$="" THEN INPUT A$
50 IF A$="S" THEN STOP
60 LET X=16*CODE A$+CODE A$(2)
-475
70 POKE S,X
80 LET Y=Y+X
90 LET C=C+1
100 LET S=S+1
110 PRINT A$( TO 2); " ";
120 LET A$=A$(3 TO )
130 IF C=5 THEN GOSUB 0150
140 GOTO 0040
150 PRINT TAB 25;Y
160 LET Y=0
170 LET C=0
180 PRINT S;TAB 8;
190 RETURN
```

Program 3.

```
100 LET F=16508
20 LET L=764
30 POKE 16444,F-256+INT (F/256)
40 POKE 16445,INT (F/256)
50 POKE 16446,L-256+INT (L/256)
60 POKE 16447,INT (L/256)
70 RAND USR 16767
80 RAND USR 16522
90 LET ZX81=PEEK 16388+256+PEE
16389+153
100 FAST
110 FOR N=0 TO 21
120 PRINT "
130 NEXT N
140 PRINT AT 1,1; "
150 PRINT AT 3,6; "ZX81=";ZX81;A
T 5,6; "PEEK 16507=";PEEK 16507;A
T 7,6; "PEEK 16508=";PEEK 16508
160 PRINT AT 10,1; "
170 PRINT AT 12,6; " RAND USR (Z
X81+2)
180 PRINT AT 15,1; "
190 PRINT AT 17,6; "GOTO 10222"
200 FAST
210 PRINT AT 0,18; "
220 PRINT AT 2,0; "1.DN CODE--LINE:AD
```

```
DR;" AT 4,0; "2.DNBASIC--" AT 5,0
"3.LN ADDR--LINE" AT 8,0; "4.AE
M EXP--LINE BYTE ADDR" AT 10,0
"5.UP CODE ADDR BYTE" AT 12,0
"6.UP BASIC 1ST LAST" AT 14,0
"7.BLK DEL 1ST LAST INCR NEUL
"8.RENUM 0 1ST LAST INCR NEUL
"9.AT 18,0; "9.RENUM.C
INCR NEUL AT 21,0; "ENTER NO."
310 INPUT N
320 PRINT AT 2+N,1; "
330 FOR M=1 TO 4
340 INPUT Z
350 PRINT AT 2+N,5+M+5; "
AT 2+N,6+M+4;Z
360 POKE 16442+M+2,Z-256+INT (Z
/256)
370 POKE 16443+M+2,INT (Z/256)
380 NEXT M
390 CLS
400 IF N=2 OR N=9 THEN RAND USR
(ZX81+590)
410 IF N=1 OR N=2 OR N=9 THEN R
AND USR (ZX81+535)
420 LET ZX82=USR (ZX81+2+N-2)
430 IF N=5 OR N=6 OR N=9 THEN R
AND USR (ZX81+501)
440 IF N=3 THEN PRINT "ADDRESS3
OF LINE "PEEK 16444+256+PEEK 16
445;" IS ";ZX82
450 REM
460 REM
```

Program 4.

```
2 SAVE "TOOLKIT"
100 FAST
200 POKE 16444,10222-256+INT (1
0222/256)
210 POKE 16445,INT (10222/256)
220 POKE 16446,10239-256+INT (1
```

```
0239/256)
230 POKE 16447,INT (10239/256)
240 RAND USR (ZX81+10)
250 POKE 16444,1
260 POKE 16445,0
270 LET N=7
280 GOTO 10235
```

appropriate address if you are resuming after a break — and enter the hex codes listed in figure 1. You may enter any number of bytes at a time, and the screen display should tally with figure 2. If you make a mistake input "S", which will stop the program. To recommence, simply Goto 10, and enter your new starting address.

When this is complete — and you should have Saved your program at several stages — Save the program thus far.

If your ZX-81 has the "unimproved" Rom, identified by Print Peek 5404 giving 253, then you must enter the five Pokes in fig.2 before Saving and proceeding further.

Type in program 3. This is most of the Basic required to transfer the code to upper memory and provide the operating menu. When you have Saved this, Run it. You should now have the menu on screen, so select 8 — Renumber — and enter the numbers 290, 460, 1, and 10222 in response to the prompts.

Almost immediately, the menu lines will be renumbered from A222 to A239 — 10222 to 10239.

If all is well, type in program 4, the remaining Basic, and Save the program again. You are now ready to Save this, the final completed program in auto-run mode. To do this simply Run the program whilst recording in the usual way. When the program has Saved, you will find that the Basic program area has been cleared and the menu lines are now stored above Ramtop where they will remain until required. Similarly, this occurs when you Load the auto-run version.

The Invisible Toolkit is now ready to use, so refer to the instructions and try it out.

Using the toolkit is easy. After Loading,

RAND USR (ZX81 + 2)

retrieves the Basic from above Ramtop, and Goto 10222 displays the menu. The first

(continued on next page)

(continued from previous page)

prompt requires you to input your selection — 1 to 9 — and the next four prompts expect numerical input, for example line number, Address — or any number if the menu indicates a blank.

When you wish to Load another program

on which to use the toolkit, there are a couple of things to remember. First make sure that the menu lines are back above Ramtop — if not, use Copy Basic to do so. Then note down the contents of locations 16507 and 16508, as you will need to restore them after Loading.

Also, the variable ZX-81 will have to be

restored after Loading — also after Run and Clear, which are best avoided.

After you have LOADED the program you intend to work on, and restored the system variables 16507 and 16508, and ZX-81

RAND USR (ZX81 + 2)

and away you go!

Figure 1.

Figure 1.							16794	22	04	40	C5	E5	=526	17044	CD	BB	05	42	4B	=536
16514	00	00	D1	00	53	=292	16799	2A	02	40	A7	ED	=512	17049	C5	CD	20	15	CD	=660
16512	00	76	76	ED	5B	=564	16804	42	22	02	40	ED	=403	17054	F0	13	C1	E1	E5	=914
16524	04	40	ED	53	7B	=511	16809	73	36	40	2A	36	=329	17059	C5	23	77	23	73	=501
16529	40	21	01	40	CD	=495	16814	40	A7	ED	42	F9	=783	17064	23	72	23	71	23	=332
16534	A5	40	D5	3E	02	=506	16819	09	ED	5B	7B	40	=524	17069	70	C1	E1	2B	10	=597
16539	CD	A5	40	CD	AD	=812	16824	EB	A7	ED	52	44	=789	17074	02	10	93	16	04	=199
16544	40	3D	20	F7	C9	=605	16829	4D	2A	36	40	ED	=474	17079	E5	D5	CD	20	15	=700
16549	23	4E	23	46	EB	=453	16834	73	5D	40	ED	5B	=600	17084	EF	C0	34	01	0A	=494
16554	09	EB	C9	EB	FD	=933	16839	5D	40	ED	B0	E1	=795	17089	00	CD	20	15	EF	=497
16559	E1	C1	C5	71	23	=763	16844	C1	D1	D5	EB	ED	=1087	17094	C1	05	24	C2	E1	=653
16564	70	EB	FD	E9	2A	=875	16849	80	E1	D1	C1	C9	=1004	17099	04	E0	01	03	E2	=458
16569	3C	40	CD	D8	09	=554	16854	10	04	10	1E	10	=106	17104	34	CD	A7	0E	C5	=635
16574	E5	2A	3E	40	23	=432	16859	12	CD	00	40	C5	=668	17109	CD	CD	15	C1	D1	=833
16579	CD	D8	09	D1	CD	=844	16864	CD	00	0A	2A	29	=394	17114	E1	C6	1C	77	2B	=613
16584	17	0A	C9	2A	04	=260	16869	40	A7	C1	ED	42	=727	17119	15	20	D5	10	96	=440
16589	40	E5	2A	3E	40	=461	16874	22	29	40	C9	FD	=593	17124	10	CD	2A	7B	40	=458
16594	10	13	10	F5	21	=345	16879	CB	01	EE	FD	CB	=898	17129	22	46	40	2A	04	=214
16599	10	27	22	2B	40	=196	16884	01	A6	10	52	FD	=526	17134	40	FD	CB	01	6E	=631
16604	2A	04	40	E5	CD	=544	16889	CB	01	AE	FD	CB	=834	17139	20	10	00	00	00	=56
16609	8B	05	EB	CD	D8	=848	16894	01	A6	21	7D	40	=389	17144	00	00	00	2A	3E	=104
16614	09	E5	2A	7B	40	=467	16899	FD	CB	01	9E	E5	=844	17149	40	23	CD	D8	09	=529
16619	ED	5B	04	40	CD	=601	16904	23	23	23	23	7E	=266	17154	22	46	40	2A	3C	=270
16624	17	0A	E1	E5	C5	=684	16909	FE	FA	20	04	FD	=793	17159	40	CD	D8	09	00	=494
16629	2B	CD	9E	09	C1	=608	16914	CB	01	DE	7E	FE	=806	17164	00	ED	4B	40	40	=440
16634	D1	E1	C5	ED	B0	=1044	16919	76	C1	23	20	1F	=417	17169	ED	5B	42	40	C5	=655
16639	C1	C5	2A	02	40	=408	16924	20	C5	7E	CD	04	=751	17174	D5	D1	C1	7A	FE	=991
16644	09	22	02	40	ED	=346	16929	07	7E	FE	EC	20	=663	17179	20	30	03	11	0F	=131
16649	73	36	40	2A	36	=329	16934	25	FE	ED	20	21	=601	17184	27	72	23	73	2B	=346
16654	40	09	F9	A7	ED	=726	16939	FE	F7	20	1D	FD	=823	17189	EB	09	EB	C5	D5	=889
16659	42	ED	5B	04	40	=462	16944	CB	01	5E	23	20	=365	17194	CD	F2	09	EB	ED	=928
16664	EB	A7	ED	52	44	=789	16949	DF	E1	CD	F2	09	=904	17199	5B	46	40	A7	E5	=621
16669	4D	ED	5B	7B	40	=592	16954	EB	ED	5B	0C	40	=639	17204	ED	52	E1	20	DE	=798
16674	2A	04	40	ED	53	=430	16959	A7	E5	ED	52	E1	=940	17209	D1	D1	FD	CB	01	=875
16679	04	40	1B	2B	ED	=375	16964	20	04	10	BB	10	=279	17214	E6	FD	CB	01	6E	=797
16684	8B	10	34	2A	3C	=362	16969	5B	10	3C	23	23	=336	17219	C0	10	9E	21	FE	=661
16689	40	CD	D8	09	44	=562	16974	23	23	23	E5	7E	=460	17224	37	CD	D8	09	23	=504
16694	4D	C9	ED	4B	3E	=632	16979	CD	B4	07	20	2D	=469	17229	23	5E	23	56	13	=269
16699	40	2A	40	40	2B	=277	16984	23	7E	FE	76	20	=565	17234	13	72	2B	73	19	=316
16704	E5	C5	CD	9E	09	=798	16989	27	E1	E5	FD	CB	=949	17239	2B	01	02	00	C5	=243
16709	C1	E1	C5	41	3E	=742	16994	01	06	20	20	23	=211	17244	E5	CD	9E	09	D1	=810
16714	1B	23	77	10	FC	=449	16999	7E	23	5E	23	5B	=376	17249	C1	21	7B	40	ED	=650
16719	10	12	10	01	10	=219	17004	23	4E	23	46	CD	=423	17254	80	C9	CD	23	0F	=632
16724	01	10	D0	10	DF	=616	17009	0D	15	C5	E1	CD	=709	17259	21	FE	27	CD	D8	=747
16729	10	24	10	2B	10	=151	17014	D0	09	10	02	10	=275	17264	09	23	23	5E	23	=208
16734	77	10	77	10	77	=405	17019	8A	E5	C1	E1	2B	=876	17269	56	D5	1B	1B	72	=467
16739	2A	3C	40	CD	D8	=587	17024	70	2B	71	10	B1	=469	17274	2B	73	D1	D5	19	=605
16744	09	C1	20	0B	23	=280	17029	D1	10	AE	FD	CB	=863	17279	7E	FE	EA	20	07	=653
16749	23	5E	23	56	EB	=485	17034	01	06	20	5B	C9	=432	17284	D1	2B	72	2B	73	=524
16754	09	EB	72	2B	73	=516	17039	2B	56	2B	5E	EB	=501	17289	CF	13	D1	01	02	=438
16759	2A	29	40	09	22	=190								17294	00	C5	11	7C	40	=402
16764	29	40	C9	2A	3C	=408								17299	ED	5B	23	C1	E5	=878
16769	40	ED	4B	3E	40	=502								17304	09	EB	E1	CD	60	=770
16774	10	03	CD	8B	40	=480								17309	0A	C9	2A	04	40	=321
16779	C5	D5	E5	CD	C5	=1041								17314	7E	FE	20	30	09	=477
16784	0E	2A	04	40	22	=158								17319	2A	7B	40	2B	7E	=390
16789	7B	40	A7	ED	42	=657								17324	FE	76	20	02	CF	=621
16794														17329	1C	C9				

Figure 2.

POKE 17051,20
POKE 17054,244
POKE 17082,20
POKE 17091,20
POKE 17110,201

Figure 2.

POKE 17051,28
POKE 17054,244
POKE 17082,20
POKE 17091,20
POKE 17110,201